SEGMENTATION OF AUTOMATICALLY TRANSCRIBED BROADCAST NEWS TEXT

P. van Mulbregt, I. Carp, L. Gillick, S. Lowe and J. Yamron

Dragon Systems, Inc.
320 Nevada Street
Newton, MA 02460

ABSTRACT

Expertise in the automatic transcription of broadcast speech has progressed to the point of being able to use the resulting transcripts for information retrieval purposes. In this paper, we describe the Segmentation system used by Dragon Systems in the Segmentation task of the 1998 TDT evaluation, highlighting improvements made since the September 1998 dryrun. Segmentation of closed-caption and human transcripts of news is contrasted with the results of segmenting the ASR transcripts. This will be followed by a discussion of the metric, in particular how the value of the segmentation metric relates to the value of the tracking metric, when the these latter two tasks are performed on automatically segmented ASR text, rather than the ASR text with correctly marked boundaries.

1. INTRODUCTION

In [1], we introduced a new approach to text segmentation and topic tracking, one based on HMMs and classical language modeling. In that paper we applied the method to segment text from the Topic Detection and Tracking (TDT) Pilot Study Corpus, made up of Reuters newswire and manually transcribed CNN news stories. Dragon's approach is to build a network, where each node corresponds to a unigram language model and sentence/utterance of text, and find the best path through this network. The score of each node is the score of its sentence in its language model. The transition penalty is zero for transitions between consecutive sentences with the same language model, and is a tunable "switch penalty" for transitions to a different language model. The language models are built by automatically clustering stories into N clusters, and then creating a unigram language model from each cluster, smoothed with a global unigram model. In [2], we discussed segmentation results on a subset of the TDT2 corpus, as well as tracking results on the automatically segmented text in comparison tracking on the correctly segmented text. Here we discuss improvements to the segmenter arising from modeling non-news text in the ASR stream and making use of a larger training set, as well as giving results in the 1998 TDT-2 evaluation.

2. THE TDT2 CORPUS

The TDT2 Corpus consists of about 60,000 news stories from television, radio, and newswire, collected by the Linguistic Data Consortium (LDC) over the period January 1998 through June 1998. The broadcast portion includes both closed-caption and automatic transcriptions of entire shows from the Cable News Network (CNN), American Broadcasting Company (ABC), Public Radio International (PRI), and Voice of America (VOA), one show from each of the first three source, and 3 distinct shows from VOA. The newswire component of the corpus was collected from the Associated Press Worldstream (APW) and the New York Times News Service(NYT). The television and radio shows were automatically transcribed as described in [3].

3. THE SEGMENTER

Suppose that there are k topics, each a collection of text documents, $T^{(1)}$, $T^{(2)}$, ..., $T^{(k)}$. (In this paper, use of the term topic for a collection should not be confused with the use of the same term used as part of the Topic Tracking task of TDT.) There is a language model associated with each topic $T^{(i)}$, $1 \leq i \leq k$, with which one can calculate the probability of any sequence of words. In addition, there are transition probabilities among the topics, including a probability for each topic to transition to itself (the "self-loop" probability), which implicitly specifies an expected duration for that topic. Given a text stream, a probability can be attached to any particular hypothesis about the sequence and segmentation of topics in the following way:

1. Transition from the start state to the first topic and accumulate a transition probability.
2. Stay in topic for a certain number of words or sentences, and, given the current topic, accumulate a self-loop probability and a language model probability for each.
3. Transition to a new topic, accumulate the transition probability, and go back to step 2.

A search for the best hypothesis and corresponding segmentation can be done using standard HMM and speech recognition techniques. This segmentation does provide a label for each segment, namely the topic to which that segment was assigned in the best path, but the label may not have much meaning to a human.

3.1. Constructing the Topic Language Models

The topic language models used by the segmenter were built from the newswire and the automatically transcribed broadcasts from the January through April data.

This totaled about 15 million words spread across about 48000 stories of average length 310 words, though the average length varied from a low of 129 for CNN, to a high of 850 for the New York Times. A global unigram model consisting of 60,000 words was built from this data.

Topic clusters were constructed by automatically clustering the stories in the training data. This clustering was done using a

multi-pass k-means algorithm described in [1]. In order to prevent very common words and punctuation symbols from dominating the computation, we introduced a stop list containing 112 entries. These words did not participate in the computation of the distance measure. Removing these words from the vocabulary meant that approximately half the words in the text stream were not scored.

A topic language model was built from each cluster. We chose to model each topic using unigram statistics only. These unigram models were smoothed versions of the raw unigram models generated from the clusters. Smoothing each model consisted of performing absolute discounting followed by backoff [4] to the global unigram model; in other words, a small fixed count (about .5) was subtracted from the non-zero raw frequencies, and the liberated counts were redistributed to the rest of the words in the model in proportion to the global unigram distribution built from the training data. The raw cluster unigrams were quite sparse, typically containing occurrences of only 6,000 distinct words from the training list of 60,000 words. Words on the stop list were removed from the models.

We will frequently refer to these topic language models as *background topics* or *background models.* Note that these models could have been built from any news sources — the idea is for the clusters to represent all of English news discourse — and not be source dependent.

4. SEGMENTATION RESULTS ON THE 1998 TDT2 EVALUATION DATA

The TDT-2 evaluation was conducted on the automatically transcribed portion (ASR) of the TDT2 Corpus, taken from the months of May and June. This collection comprised 384 shows, 6000 stories, 2.2 million words. About 60% of the shows are from CNN.

The ASR output has breaks marked; typically silence, music or speech with a music background, and were used to identify possible story transition times.

Decoding of text was done by using a speech recognizer with k underlying "single node" models (corresponding to the topics), each of which was represented by a unigram model as described above. The text was scored against these models one *frame* at a time—a frame corresponding, in these experiments, to the words between recognizer breaks. The topic-topic transition penalties were folded into a single number, the topic-switch penalty, which was imposed whenever the topic changed between segments.

There is one important parameter in the system requiring tuning, the topic switch penalty. The switch penalty depends on the broadcast source, and the granularity of the background models, and should be robust to slight perturbations of the models. The actual tuning was accomplished by training a model with k topics on all except a small held-out set of data, tuning the switch penalty on this help-out set, and using this switch penalty for models with k topics built on all of the allowable training data. There are no other parameters to tune except the search beam width, which was set large enough to avoid search errors.

The switch penalty is the only parameter depending on the broadcast source, and as such the models can be used on any source as long as the average document length is known.

The segmenter was first run on the ASR portion of the TDT2 Corpus. Then contrasting runs were made on the Closed Caption and FDCH Transcripts text of the corpus.

All results are reported using the CSeg metric for measuring the quality of a segmentation. The metric takes the form

$$C_{Seg} = P_{Seg} * P_{Miss} + (1 - P_{Seg}) * P_{FalseAlarm}$$

where P_{Miss} and $P_{FalseAlarm}$ are computed with a window width of 50 words and P_{Seg} is the *a priori* probability of a segment boundary being within the window length. For the ASR portion of the TDT2 corpus, P_{Seg} is 0.3 corresponding to an average story-length of $50/0.3 \approx 165$. So $C_{Seg} = 0.3 * P_{Miss} + 0.7 * P_{FalseAlarm}$ The full details are in [5].

4.1. The Main Evaluation

Show	P(Miss)	P(FA)	CSeg
ABC_WNT	0.3454	0.0888	0.1658
CNN_HDL	0.3094	0.1022	0.1644
PRI_TWD	0.3056	0.0670	0.1386
VOA_ENG	0.3333	0.0772	0.1540
VOA_TDY	0.3210	0.0695	0.1449
VOA_WRP	0.3448	0.0635	0.1479
Overall	0.3183	0.0835	0.1539

Table 1: Official Segmentation Performance on ASR data broken out by source.

There is a little variation across source. Our algorithm tends to work best on material which is mostly content, with few segues and fillers.

There does seem to be a bias in the metric towards undergenerating segments, which accounts for the disparity between misses and false alarms. Typically the optimum number of segments to generate to minimize C_{Seg} is somewhere between 65% and 80% of the true number of segments.

4.2. Contrast - FDCH Transcripts and Closed Captioned Text

Source Condition	CSeg	CSeg for ABC
ASR	0.1579	0.1723
FDCH Transcripts	0.1149	0.1515
Closed Captioned	0.1138	0.1356

Table 2: Overall and ABC Segmentation Performance according to source condition

Closed Caption text is available for all the sources, and these were also tested on. For ABC, the FDCH Transcripts were also available, which gave a comparison between human transcriptions and automatic transcriptions. The official FDCH test substituted FDCH transcripts for Closed Caption transcripts for ABC, and left the other sources intact.

The system perfoms better on closed caption and transcripts than on ASR text. There appear to be differences between the closed caption and transcripts on the one source for which we have all three, about 10% improvement for the true transcript, and a further 10% for using the closed captioning.

4.3. Contrast - Deferral Period

The approach we have taken requires the whole text stream to be processed first. Then a traceback occurs and the boundaries determined by the segmentation are marked. It is possible to restrict the

Deferral Period (words)	100	1000	10000
CSeg	0.1852	0.1580	0.1579

Table 3: Segmentation Performance on ASR text broken out by Deferral Period

segmenter so that at any position in the text stream, it only looks ahead a fixed number of words and must output a decision immediately as to whether or not a segment boundary occurs at this position. This capability might be desired for an "online system". Earlier experiments performed on the TDT2 corpus suggest that there is a small degradation in the method as the length of the look ahead period decreases, but that if the segmenter is allowed to look ahead twice the average story length, then it almost never changes its mind. Our system's performance is the same with deferral periods of 10000 or 1000, with about 20% degradation for a deferral period of 100.

4.4. Varying the Number of Background Models

The choice of number of topic models is somewhat arbitrary. In earlier experiments [2], we had used 50, 100 or 250 topic models, with 250 showing a slight win over 100 in some conditions, both much better than 50. With the doubling of the available training data, it seemed possible to build models with more topics and models with 250 and 400 topics were built.

While a larger number of models will provide more resolution, it also results in clusters with less data and perhaps less robust estimates for the unigram probabilities. However, the amount of training data used seems appropriate for the large models. In tuning experiments on broadcasts covering a short time frame immediately following the training period, the large models did indeed seem to perform better than the smaller models. On the evaluation test data, whicb covered a longer time frame, the 250 and 400 models performed comparably. It is possible that some of the stories immediately following the training period are well represented by stories in the training data (they may even be in the training data) and the larger models have an advantage, but the large models lose that sharper edge as time elapses.

4.5. The Effect of the Miscellaneous Models

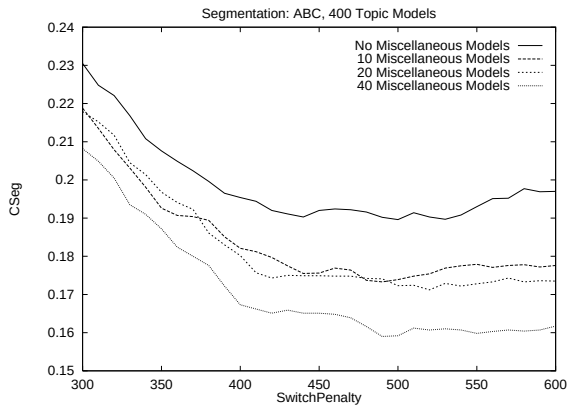

Figure 1: Varying the Number of Miscellaneous Models

The TDT ASR text contains advertisements and sports as well as news stories, and non-news stories are marked as "Miscellaneous" by the LDC. Up to 10% of the data falls into this category. These stories are not scored in the TDT2 evaluation, but their presence could be affecting segment boundary placements close by. Hence we decided to build some number of "miscellaneous" models, and add these to the collection of topic models. The miscellaneous text was automatically clustered into 10, 20 or 40 clusters, and unigram langauge models built from these clusters.

In Figure 1, the effect of the miscellaneous models is shown as the switch penalty is swept out to view a full graph. More models appear to offer an improvement, and certainly 40 miscellaneous models is a big improvement over no miscellaneous models. Again there may be material (in this case, advertisements) that appears in both the training data and the evaluation data, giving a boost to the models built from more miscellaneuous clusters.

4.6. Multi-Node Models for the Segmenter

In order to address the first two problems with the segmenter, we introduced two new language models to model the beginning and end of a story. The segmenter was optionally allowed to spend one unit of text (the text between recognizer breaks) starting a story, and one unit ending a story, and incur a penalty for doing so.

The story-beginning (story-end) model was trained by taking the first (last) unit of text from all the stories in the training data and building unigram language models. This does include parts of stories that a human reader would probably consider content of the story rather than filler or segue - a future version of the segmenter will train these models directly.

However, these models showed no improvement over the single-node models. A mixture model (or story-beginning and topic) might be more appropriate as way of using the story position information to better the segmentation.

4.7. Errors

The types of errors made by the segmenter fall into the following categories:

- Failure to distinguish a boundary between successive stories because they were assigned to the same background topic. This didn't seem to be a large source of errors, but the effect can be reduced by increasing the number of background models.
- Failure to accurately position boundaries relative to "broadcast filler", such as, "More news after this." This is a weakness of a system that does not model story structure.
- Splitting of stories at internal topic shifts. This "problem" actually goes to the heart of what it is we are trying to accomplish, and it is not clear that this is always undesirable behavior.
- Oscillation of topic in stories not well-modeled by the background topics. This might be solved by using models with better discrimination, such as bigram models, or models that adaptively train so as to stay current.
- Oscillation of topic in long stories. In some cases knowledge of the structures of the show (some initial one sentence headlines, a very long story somewhere near the middle of the show), could be used to make a better determination of the boundaries.

- A portion of the story boundaries do not occur at breaks as output by the automatic speech recognizer, and as such can never be matched exactly. This might be solved by allowing a break after every word, which unfortunately leads to an explosion in the size of the search space required, as well as losing some of the robustness that comes from the accumulating language model scores for groups of words. A better approach may be to train a model for which words are more likely to end a story and allow breaks whenever these words appear in the text stream.

5. EFFECT OF SEGMENTATION ON THE TDT TRACKING TASK

The topic tracker is an adaptation of the segmenter, and is described in [1, 2]. The Tracking task is to take some initial number of stories N_t, usually between 1 and 16, and find all the subsequent stories on the same topic. The LDC has provided topics, and labelled every story as being on or off each topic. Stories may belong to multiple topics. The usual tracking task has the story boundaries provided, but one contrast is to use boundaries provided by an automatical segmentation method. In this experiment, we used the evaluation segmentations.

At very low false alarm rates, tracking on automatically segmented ASR data is comparable to tracking on transcripts, but the performance difference becomes greater at low miss rates. However, since the segmentations were chosen in order to minimize C_{Seg}, they tend to have fewer boundaries than the correct segmentations. The tracker used in the evaluation was then used to produce the results shown in Figure 2, with $N_t = 4$.

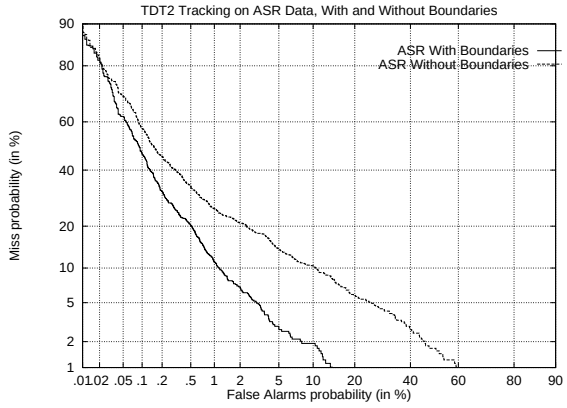

Figure 2: The Effect on the Tracker of using Automatically Generated Segmentations

In order to investigate the dependence of the Tracking task upon the quality of the segmentation, a range of segmentations was produced, and then used as input to a simplified version of the evaluation tracker. The ratio of (number of hypothesised bundaries) to (number of correct boundaries) varied from 0.6 to 2.0. The resulting DET curves are shown in Figure 3.

The minimum for the Tracking task appears to be occur at a ratio of 1.0 or higher. The minimum for the Segmentation task appears to be occur at a ratio between 0.6 and 0.8. Minimizing the segmentation metric CSeg lead to segmentations which are not as good for the follow-on task of Tracking. The DET curve for

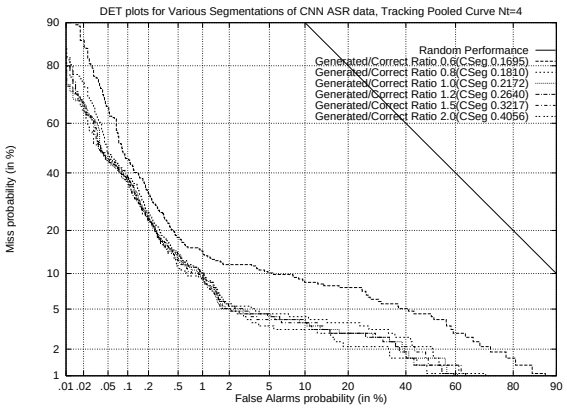

Figure 3: Tracking DET plot for $N_t = 4$ for various ratios of Generated/Correct Boundaries, CNN ASR Data only

the *best* segmentation of ASR CNN data (as measured by CSeg) is the only DET curve noticeably *worse* than all the others. A similar situation exists for the relationship between the Detection and Segmentation tasks, namely that good segmentations as measured by CSeg do not lead to good detection results as measured by CTrack. This suggests that the Tracking and Detection tasks are resilient to the quality of the segmentation, as long as there are more hypothesised boundaries than actual boundaries.

6. REFERENCES

[1] J.P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt, "A Hidden Markov Model Approach to Text Segmentation and Event Tracking," *Proceedings ICASSP-98, Seattle, May 1998*

[2] P. van Mulbregt, J.P. Yamron, I. Carp, L. Gillick, and S. Lowe, "Text Segmentation and Topic Tracking on Broadcast News via a Hidden Markov model approach." *Proceedings ICSLP-98, Sydney, December 1998*

[3] L. Gillick, Yoshiko Ito, Linda Manganaro, Michael Newman, Francesco Scattone, S. Wegmann, Jon Yamron, Puming Zhan, " Dragon Systems' Automatic Transcription of New TDT Corpus," *Proceedings of Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, Feb 1998.

[4] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400–401, March, 1987.

[5] "The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan" which may obtained from the URL <http://www.nist.gov/speech/tdt98/tdt98.htm>